Dynamic Covariance Scaling for Robust Map Optimization

Pratik Agarwal Gian Diego Tipaldi Luciano Spinello Cyrill Stachniss Wolfram Burgard

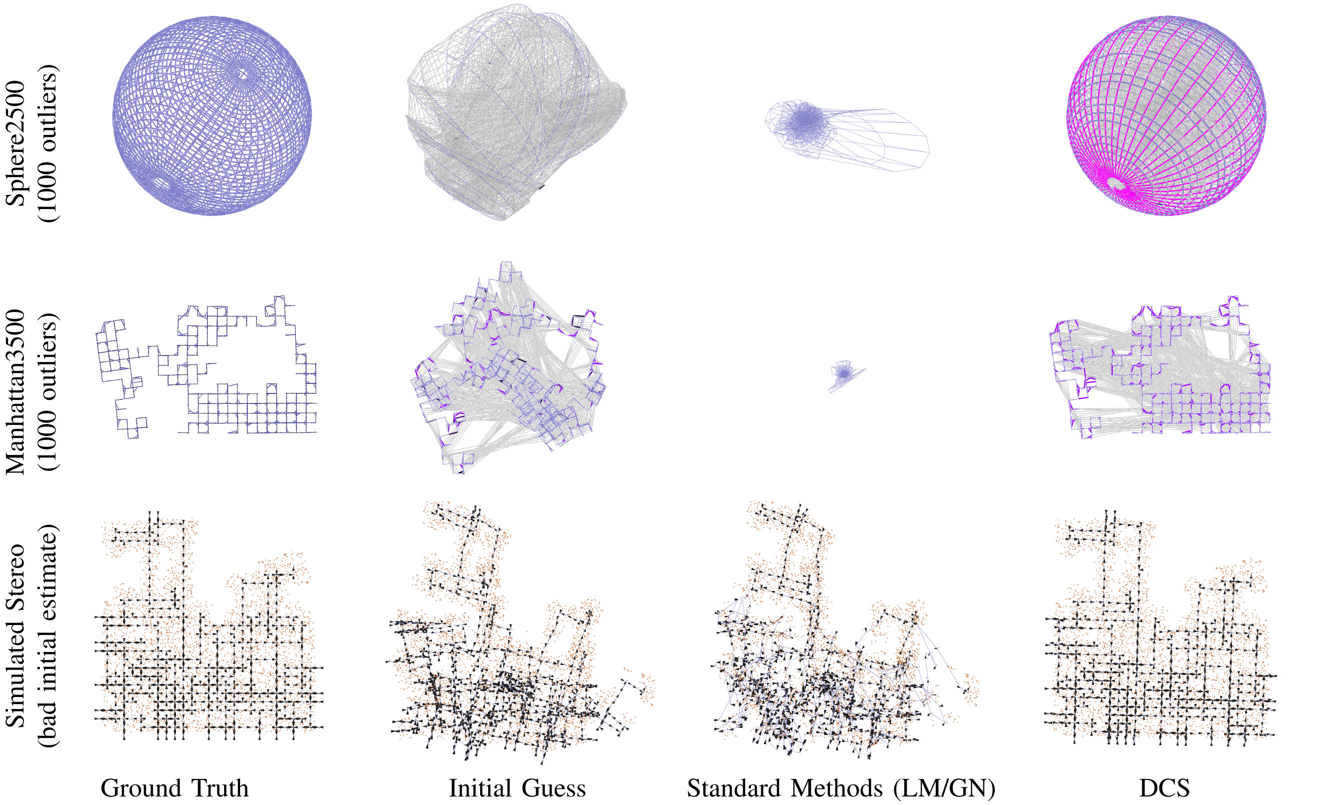


Fig. 1. The figure compares the performance of Dynamic Covariance Scaling (DCS) in the presence of outliers and bad initialization. The top two rows show DCS converging to the correct solution in the presence of outliers for the sphere dataset (top row) and Manhattan3500 (middle row). The bottom row shows DCS converging to the correct solution for a simulated outlier free dataset initialized with odometry measurements. Here, a robot equipped with a stereo camera moves in a grid world and observes point features. Levenberg-Marquardt (LM) fails to compute the optimal solution even after 100 iterations, while DCS is able to obtain a close to ground truth solution within 15 iterations. Using Gauss-Newton (GN) and Dog-Leg results in numerical issues for the stereo example. The third column shows that in all these cases, standard methods fail to reach the optimum solution.

***Abstract*—Developing the perfect SLAM front-end that produces graphs which are free of outliers is hard to achieve due to perceptual aliasing. Converging to the correct solution is challenging for non-linear error minimization SLAM techniques even in the absence of outliers, if the initial guess is far away from the correct solution. Therefore, optimization back-ends need to be resilient to outliers resulting from an imperfect front-end as well as be robust to bad initialization. In this paper, we present dynamic covariance scaling, a novel approach for effective optimization of constraint networks under the presence of outliers and bad initial guess. The key idea is to use a robust function that generalizes classical gating and down-weights outliers without compromising convergence speed. Compared to recently published state-of-the-art methods, we obtain a substantial speed-up without increasing overheads.**

I. Introduction

Building maps with mobile robots is a key prerequisite for several robotics applications. As a result, a large variety of SLAM approaches have been presented in the robotics community over the last decades [5, 4, 19, 12, 7]. One intuitive way of formulating the SLAM problem is to use a graph. The nodes in this graph represent the poses of the robot and position of features observed at different points in time, while the edges model constraints between them. The edges are obtained from observations of the environment or from motion carried out by the robot. Once such a graph is constructed, the map can be computed by optimization techniques. The solution is the configuration of the nodes that is best explained by the measurements.

Most approaches assume that the constraints are affected by noise but no outliers (false positives) are present, i.e., there are no constraints that identify actually different places as

All authors are with the University of Freiburg, Institute of Computer Science, 79110 Freiburg, Germany

This work has been partially supported by BMBF, contract number 13EZ1129B-iView and European Commission under contract numbers ERC-267686-LifeNav, FP7-600890-ROVINA and FP7-248873-RADHAR.

being the same one. This corresponds to the assumption of having a perfect SLAM front-end. In traditional methods, a single error in the data association often leads to inconsistent maps. Generating outlier-free graphs in the front-end, however, is very challenging, especially in environments showing self-similar structures [22, 17, 3]. Thus, having the capability to identify and to reject wrong data associations is essential for robustly building large scale maps without user intervention. Recent work on graph-based SLAM addressed the issue and there are now methods that can handle a large number of outliers [21, 18, 14].

The initial configuration of the graph to be optimized can also have a strong impact on the final result as the error minimization procedure may get stuck in a local minima. This holds for pose-graph SLAM as well as for graphs that contains robot poses and features. The sensor characteristics and the choice of the observation function has a strong impact on the convergence properties.

The contribution of this paper is a novel approach, namely Dynamic Covariance scaling (DCS) which is resilient to outliers and bad initial guess as shown in Figure 1. At the same time DCS avoids an increase in execution time. Our work stems from the analysis of a recently introduced robust back-end based on switchable constraints (SC) [21] and uses a robust function that generalizes classical gating by dynamically scaling the covariance. Compared to state-of-the-art approaches in robust SLAM back-ends, our strategy has a reduced computational overhead and typically has better convergence. The proposed function shares common grounds with existing robust M-estimators. We also evaluate DCS in situations considered by Grisetti et al. [6] on SLAM graphs with bad initial guess. We show that DCS offers similar convergence properties as compared to [6], without requiring any condensed measurements, partitioning of the graph, re-initialization, or similar.

II. Related Work

Various SLAM approaches have been presented in the past. Lu and Milios [16] were the first to refine a map by globally optimizing the system of equations to reduce the error introduced by constraints. Subsequently, Gutmann and Konolige [9] proposed a system for constructing the graphs and for detecting loop closures incrementally. Since then, many approaches for minimizing the error in the constraint network have been proposed, including relaxation methods [11, 5], stochastic gradient descent and its variants [19, 8], smoothing techniques [12] and hierarchical techniques [2, 7].

The techniques presented above allow for Gaussian errors in the constraints of the pose-graphs, i.e., noisy constraints, but they cannot handle outliers, i.e., wrong loop closing constraints between physically different locations. Although SLAM front-ends (loop generation and loop validation) improved over the last years [3, 17, 22], it is not realistic to assume that the generated pose-graphs are free of outliers.

Hence researchers recently started using the back-end slam optimizer to identify outliers. For example, Sünderhauf and Protzel [21] proposed a technique that is able to switch off potential outlier constraints. The function controlling this switching behavior is computed within the SLAM back-end. Olson and Agarwal [18] recently presented a method that can deal with multi-modal constraints, by introducing a max operator. Their approach approximates the sum of Gaussian model by the currently most promising Gaussian. This allows for dealing with multi-modal constraints and rejecting outliers while maintaining computational efficiency. Latif *et al.* [14, 15] proposed RRR, which handles outliers by finding the maximum set of clustered edges, consistent with each other. The key difference of RRR to the previously described two approaches [21, 18] is that RRR rejects false edges while the other two always keep rejected edges with a low weight. Our approach is similar to [21] since we also keep rejected constraints with a small probability, but it is more principled and leads to faster convergence.

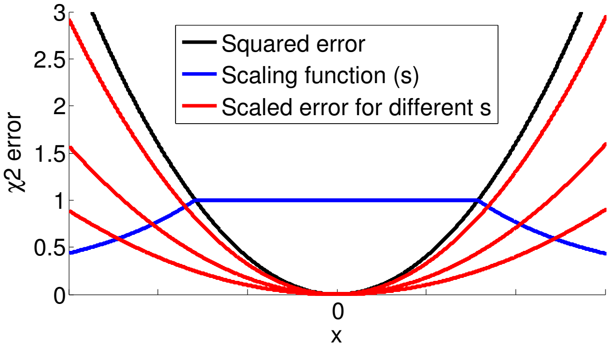


Fig. 3. Toy example of the DCS cost kernel. The scale function s has a maximum value of 1, at which DCS behaves like a normal squared error function. As the error is increased, the scaling function decreases reducing the χ^2 error.

Recently, Grisetti et al. [6] showed that bad initializations quickly lead to divergence especially in the context of non-linear models. They propose to partitions the factor graph with a divide-and-conquer approach to exploit local estimates. As shown in [6], this offers a larger convergence basin than Levenberg-Marquardt and yields convergence to the true solution in several real world and simulated environments where other state-of-the-art methods fail.

III. Dynamic Covariance Scaling

Dynamic covariance scaling or DCS [1] is a robust estimator having close resemblance to Geman-McClure [23], which prevents constraints from adding large errors. DCS handles outlier constraints by scaling their information matrix and reducing its effect on the optimizer. In this respect, DCS is similar to the switchable constraints (SC) approach [21], where outliers are not rejected but only down weighted. The difference is that SC optimizes over the scaling variables, while DCS computes them in closed form.

In the DCS formulation, the original least square SLAM [7, 10] formulation,

$$X^* = \operatorname*{argmin}_X \sum_{ij} \mathbf{e}_{ij}(X)^T \Omega_{ij} \mathbf{e}_{ij}(X) \qquad (1)$$

is augmented such that each loop closing constraint of Eq. 1 now has a scaling variable, which scales the $\chi^2_{l_{ij}}$ as follows:

$$\begin{aligned} X^* &= \underset{X}{\operatorname{argmin}} \sum_i \mathbf{e}_{i,i+1}(X)^T \Omega_{i,i+1} \mathbf{e}_{i,i+1}(X) \\ &\quad + \sum_{ij} s_{ij}^2 \underbrace{\mathbf{e}_{ij}(X)^T \Omega_{ij} \mathbf{e}_{ij}(X)}_{\chi^2_{l_{ij}}} \end{aligned} \tag{2}$$

The first summand in Eq. 2 refers to the constraints from odometry or incremental scan-matching and the second one to the loop closing constraints. The robustness is achieved by scaling each error term $\mathbf{e}_{ij}$ with s_{ij} or by scaling the information matrix Ω_{ij} with the squared of the scalar s_{ij}^2

$$\mathbf{e}_{ij}^{DCS} = \mathbf{e}_{ij}(X)^T (s_{ij}^2 \Omega_{ij}) \mathbf{e}_{ij}(X) \tag{3}$$

This reduces the confidence of outlier measurements. The scaling variable s_{ij} is computed as

$$s = \min\left(1, \frac{2\Phi}{\Phi + \chi_l^2}\right), \tag{4}$$

where Φ is a free parameter. A detailed derivation of the scaling function and an analysis of the impact of Φ can be found in [1].

In sum, we have a closed form solution for computing the scaling factor s individually for each loop closing constraint. It depends on χ_l^2, which is the original error term for each loop closing constraint. This formulation dynamically scales the information matrix of each non-incremental edge by s^2 given by Eq. 4 and thus by a factor that considers the magnitude of the current error. A gradient always exists in the direction of an edge and gradually increases in the presence of more mutually consistent constraints. The cost surface is always quadratic but the magnitude of the gradient is scaled according to s, which depends on the current error (χ_l^2) and Φ.

In practice, DCS has the effect of down-weighting constraints with large errors. Fig. 3 shows a toy example of the DCS kernel. The scaling function for a constraint remains flat when $\chi^2_{l_{ij}} \leq \Phi$. In this region, DCS behaves like a normal squared kernel without any scaling. As the error increases, DCS scales the information matrix gradually. This has the effect of the error still being squared but with reduced weight. As, $\chi^2_{l_{ij}} \rightarrow \infty$, $s \rightarrow 0$.

IV. Experiments

To support our claims that DCS is resilient both to outliers and to bad initial guess in the context of non-linear SLAM, we evaluate it on publicly available datasets. These include Manhattan3500, Intel Research Lab, City10000, Sphere2500, CityTrees10000, and Victoria Park datasets. For Manhattan3500, we considered the two different initialization procedures provided by Olson [19] and g2o [13]. The Intel Research Lab dataset is available in the g2o package [13] and the City10000, CityTrees10000, Sphere2500 datasets as well as the Victoria Park dataset were released with the iSAM package [12]. We also evaluated additional large-scale datasets such as the 36 loops of the Lincoln Lab and the five loops of the Bicocca multi-session experiment initially evaluated with RRR [14].

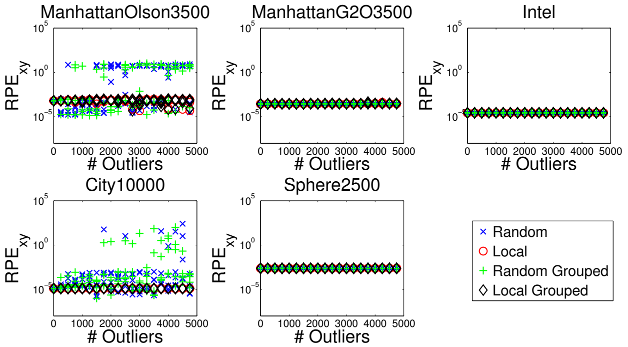


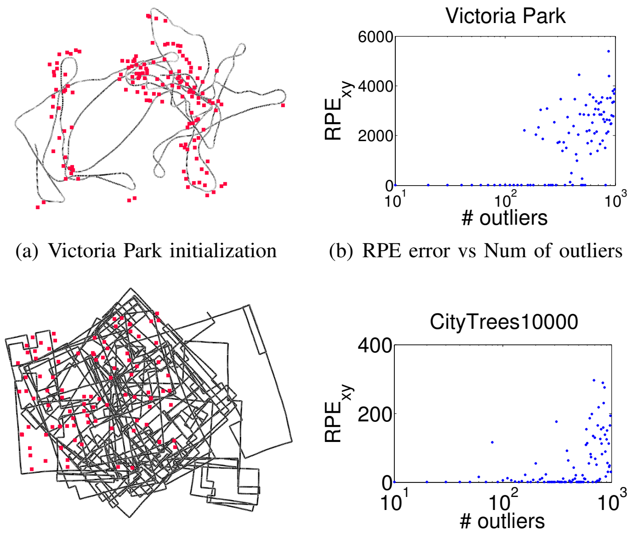
Fig. 4. Scatter plots showing the error depending on the number and type of outliers for DCS. ManhattanG2O, Intel, and Sphere2500 converge to the correct solution even with 5,000 outliers while City10000 and ManhattanOlson always converges in the case of local outliers. City10000 converges to the correct solution for up to 1,500 outliers which are not local. ManhattanOlson is more sensitive to non-local outliers.

The corrupted versions of the data sets contain both, real and simulated outliers. For simulated outliers, we used four different approaches to generate them namely "random", "local", "random grouped", and "local grouped" as described in [21]. Random outliers connect any two randomly sampled nodes in the graph. Local outliers connect random nodes that are in the vicinity of each other. For the grouped outliers, we create clusters of 10 mutually consistent outliers. We believe that randomly grouped outliers are the most realistic form of outliers as such constraints are similar to systematic errors generated due to perceptual aliasing by a front-end. The outliers are generated using the script provided in the Vertigo package [20]. For landmark datasets such as Victoria Park and CityTrees10000, we added wrong loop closures between random pairs of nodes and landmarks.

For the Bicocca and Lincoln multi-session datasets, we used the processed datasets provided by Latif et al. [14] in which loop closures are generated using a place recognition system subjected to perceptual aliasing. The Bicocca dataset uses a bag of word-based front-end while the Lincoln Lab dataset was created with a GIST-based front-end.

A. Robustness against Simulated Outliers

To show the robustness against outliers we evaluated DCS on both simulated and real outliers. First, we evaluated DCS on datasets with up to 5,000 simulated outliers. In total, we evaluated 400 graphs per dataset—100 for each of the four outlier generation strategy. Scatter plots of the resulting reprojection error (RPE) after convergence are shown in Fig. 4. As can be seen, for the ManhattanG2O, Intel and Sphere2500 datasets, DCS always converges to the correct solution. For ManhattanOlson and City10000, DCS converges in all the local outlier cases but is sensitive to the non-local outliers. City10000 fails to converge to the correct solution in some non-local cases with more than 1500 outliers. Even when ManhattanOlson does not converge, the RPE is always less than 10 and appears somewhat constant. This case is further



(a) Victoria Park initialization (b) RPE error vs Num of outliers

(c) CityTrees10000 initialization (d) RPE error vs Num of outliers

Fig. 5. Resulting RPE for Victoria Park and cityTrees10000 dataset in the presence of a varying number of outliers. Although the initialization is far from the global minimum, DCS is able to converge to the correct solution for small number of outliers.

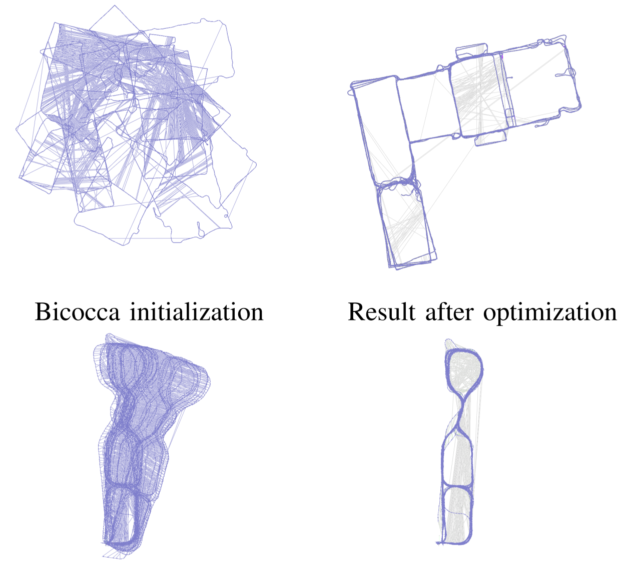
analyzed in Section IV-E.

Most landmark-based SLAM systems provide pose-to-feature range-bearing constraints and pose-to-pose constraints only for odometry. Operating on pose-to-feature constraints is more challenging for outlier rejection since there is no reliable constraints such as odometry between the feature nodes. In the previous evaluated pose graphs, every node is constrained by two odometry edges which are not subjected to being an outlier. For landmark datasets all constraints to a feature node are potential outliers and hence create large number of local minima solutions.

For landmark datasets we corrupt the outlier free Victoria Park and the CityTrees10000 dataset with up to $1,000$ random outlier constrains. The outliers are random measurements from a robot pose to a landmark. Fig. 5 shows the initialization for these two datasets and the RPE with an increasing number of outliers. As can be seen from the plots, in these two datasets, DCS is robust up to around 100 outliers and the robustness decreases as the outliers are increased thereafter. The fact that DCS is still able to optimize the Victoria Park dataset from the initialization shown for 100 random outliers is strong evidence that the method can be used in synergy with existing front-ends validation techniques for landmark based system to improve robustness.

B. Robustness against Real Outliers

Compared to the other datasets evaluated in this paper, the next two datasets contain outliers created from a front-end due to place recognition errors. The goal of these experiments is to evaluate the performance of DCS with an error prone front-end. We use the data-sets evaluated in [14] and thus also provide an informal comparison to it. Fig. 6 depicts the optimization results of DCS on the Biccoca and Lincoln Lab datasets.



Bicocca initialization Result after optimization

Lincoln Labs initialization Result after optimization

Fig. 6. Qualitative evaluation on 5 sessions of Bicocca (top) and 36 loops of Lincoln Labs (bottom) datasets that contain outliers generated by the vision system. Latif et al. [14] report a that RRR solves Bicocca in 314 s whereas DCS requires only 1.56 s to obtain the solution.

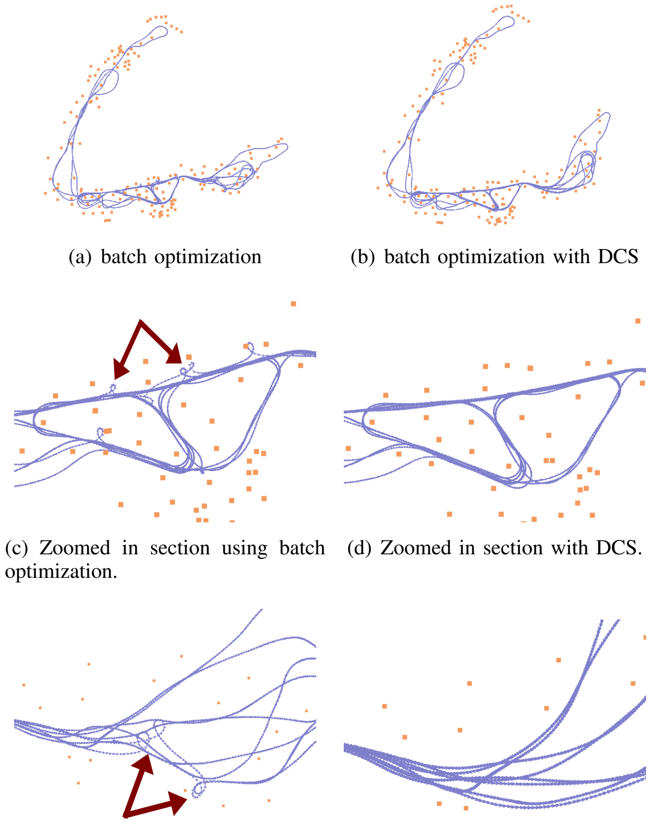
DCS takes 0.79 s (3 iterations) to optimize the Lincoln Lab dataset and 1.56 s (16 iterations) to optimize the Bicocca dataset. For the Bicocca dataset, we achieved the best result with $\Phi = 5$. By visual inspection, we can see that our solution is close to the reported correct structure in [14]. Compared to RRR, which reports a timing of 314 s for the Bicocca dataset, DCS takes only 1.56 s and thus is around two orders of magnitude faster. SC does not find the correct solution in the standard settings and requires an additional Huber robust kernel which takes 5.24 s to find the solution [14].

C. Robustness with respect to Bad Initialization

In the previous sections we showed that DCS is resilient to outliers in SLAM-graphs with different sensor modalities. The aim of this experiment is to show that DCS can reach the global optimum in outlier-free datasets where standard non-linear methods, i.e., Gauss-Newton, Levernberg-Marquardt and Dog-Leg fail due to bad initialization. The original Victoria Park dataset contains range-bearing observations of trees, which are used as point landmarks. It contains a total of 151 landmarks observed from 6.969 poses. This high pose to landmark ratio makes the problem challenging to converge for batch methods as illustrated in Figure 7.

Visually, the batch method with Gauss-Newton without DCS seems to converge to the correct solution as shown in Fig. 7(a), but a more detailed analysis reveals that this is not the case. Figures 7(c) and 7(e) show enlarged parts for the solution obtained by batch methods. Non-existing loops appears in the odometry chain, which corresponds to local minima. Figures 7(d) and 7(f) show the correct results obtained with DCS.

The total χ^2 error of the solution with DCS is 390.95 compared to $30,607.16$ with Gauss-Newton, $13,319.25$ with Dog-



(a) batch optimization (b) batch optimization with DCS

(c) Zoomed in section using batch optimization. (d) Zoomed in section with DCS.

(e) Zoomed in section using batch optimization. (f) Zoomed in section with DCS.

Fig. 7. Optimization of Victoria-Park dataset with range-bearing measurements. The batch solution without DCS converges to the wrong solution. The errors in the robot poses can be clearly seen as small loops in the odometry chain. These are not present when used with DCS. The batch solutions have a total error of $30,607.16$ compared to an error of 390.95 with DCS. Best results were obtained with $\Phi = 1$.

Leg and $87,147.58$ with Levenberg-Marquardt. The solution obtained with DCS is similar to the CM approach [6].

D. Comparison to Switchable Constraints

Experiments is this section compare the timing and convergence properties between DCS and SC. Tab. I compares the time required by DCS and SC to converge in presence of outliers. This table compares the total time taken for both these algorithms to reach the optimal solution. As can be seen from the table, DCS is faster than SC in all cases. The increase in convergence speed is most noticeable in City10000 dataset. The optimization process for both methods were stopped when the change in χ^2 was less than the set threshold. In the next section we show that the reduction of χ^2 and RPE is significantly faster and smoother for DCS compared to SC.

The next set of experiments analyze the convergence behavior of DCS and SC in the presence of 1000 randomly grouped errors, as this is the most difficult and realistic scenario. Fig. 8 plots the evolution of the RPE (top row) and the χ^2 error (bottom row) during optimization for SC (blue) and DCS (green). As can be seen from these plots, within at most 6 iterations, DCS converges while SC typically needs between 15 and 20 iterations to converge. The shapes of the plots for SC reveal a frequent increase of the RPE as well as χ^2 error. We believe this may be indicative of the fact that the Gauss-Newton quadratic approximation of the cost functions for the new optimization problem with additional switch variables in SC is not completely accurate in the neighborhood of evaluation.

For our method, the evolution of χ^2 and RPE is smooth and almost monotonous. The plots illustrate that DCS requires a smaller number of iterations and offers a faster convergence while at the same time being robust to outliers. This is also apparent from the video submitted with the paper, available at http://www.informatik.uni-freiburg.de/%7Eagarwal/videos/icra13/DCS.mp4. Note that the absolute χ^2 values for SC and DCS have to be interpreted differently since SC introduces extra switch prior constraints contributing to the overall error.

E. Degenerate cases

Experiments in this section are designed to bring out subtle differences and challenges which remain to be solved even with these state-of-art methods.

Investigations of the failure cases in ManhattanOlson found is Section IV-A reveal an interesting behavior. We analyses two specific failure cases, one with 501 and one with 4,751 random outliers. After converging, both solutions appear to have similar configuration, even though the second case is subjected to roughly ten times more outliers, shown in Fig. 9. They are locally consistent and appear to have converged to a similar local minimum. The scaling values of each false positive edge is shown in the plots in Fig. 9. The problem here is that three parts of the graph are only sparsely connected (see Fig. 9-left). By adding non-local and mutual consistent outliers, there exists configurations in which the system cannot determine all outliers correctly. SC shows a similar issue with ManhattanOlson, which the authors solved by introducing an additional robust Huber kernel at the expense of an even slower convergence [21].

The parking garage dataset is a difficult real world dataset compared to all the previous ones. This is mainly because of the sparse nature of loop closures. Each deck of the parking garage is connected by two odometry chains. SC had reported degenerate behavior with this dataset [21]. The authors argued that since only a small number of constraints connect the decks robust methods were not able to outperform non-robust methods.

DCS is able to reject outliers even in this dataset. We also added mutually consistent constraints between decks at multiple levels and compared both methods with standard parameters as shown in fig 10. We believe DCS is able to reject outliers as the gradients of odometry edges and correct loop edges outweigh those provided by the outliers.

V. Conclusion

In this paper, we presented dynamic covariance scaling (DCS), a principled method to cope with outliers and

TABLE I

OPTIMIZATION TIME NEEDED BY SC AND DCS IN THE PRESENCE OF 1000 TO 5000 OUTLIERS WITH RANDOM(R), LOCAL(L), RANDOM-GROUPED(RG) AND LOCAL-GROUPED(LG) OUTLIER GENERATION STRATEGIES.

Dataset	1000 R, L, RG, LG	2000 R, L, RG, LG	3000 R, L, RG, LG	4000 R, L, RG, LG	5000 R, L, RG, LG
ManG2O					
SC	4.70, 1.91, 3.11, 1.55	8.17, 2.93, 4.46, 2.85	10.11, 3.45, 11.89, 5.11	11.21, 2.80, 11.17, 3.32	24.53, 3.14, 15.33, 4.67
DCS	**2.09, 0.86, 1.41, 0.88**	**3.83, 1.07, 2.80, 1.00**	**5.47, 1.25, 4.24, 1.17**	**7.62, 1.44, 6.27, 1.38**	**9.29, 1.69, 8.42, 1.59**
ManOlson					
SC	14.53, 2.21, 10.65, 2.21	18.96, 2.80, 15.45, 2.71	39.34, 3.41, 39.94, 3.27	53.29, 4.69, 36.71, 4.54	67.44, 5.33, 61.16, 5.09
DCS	**4.62, 1.08, 3.40, 1.07**	**6.57, 1.35, 3.23, 1.27**	**26.21, 1.57, 20.21, 1.46**	**29.24, 1.84, 26.46, 1.71**	**16.80, 2.03, 14.00, 1.93**
Intel					
SC	0.54, 0.42, 0.51, 0.39	1.20, 0.94, 1.18, 0.94	1.60, 1.22, 1.61, 1.20	2.00, 1.52, 2.01, 1.50	2.37, 1.78, 2.44, 1.74
DCS	**0.34, 0.22, 0.31, 0.21**	**0.52, 0.31, 0.52, 0.34**	**0.69, 0.45, 0.71, 0.42**	**0.85, 0.53, 0.85, 0.50**	**1.00, 0.58, 1.08, 0.58**
City10000					
SC	47.61, 30.06, 41.11, 29.86	108.2, 33.84, 79.50, 33.52	212.8, 41.14, 134.9, 39.04	285.7, 43.82, 207.1, 40.70	389.9, 49.98, 446.5, 49.92
DCS	**10.09, 3.98, 7.88, 3.93**	**36.94, 4.80, 15.74, 4.53**	**51.60, 5.95, 34.02, 5.65**	**218.8, 6.92, 50.09, 6.44**	**262.9, 8.04, 393.2, 7.37**
Sphere2500					
SC	53.83, 11.09, 48.26, 10.62	115.5, 14.88, 108.9, 16.03	240.1, 24.10, 170.3, 18.55	218.7, 30.78, 230.2, 57.22	310.7, 67.53, 281.8, 63.37
DCS	**19.52, 7.83, 16.84, 7.51**	**42.52, 9.22, 38.39, 9.02**	**50.58, 10.40, 50.32, 9.94**	**66.51, 11.31, 69.39, 11.36**	**90.12, 12.35, 97.07, 11.97**

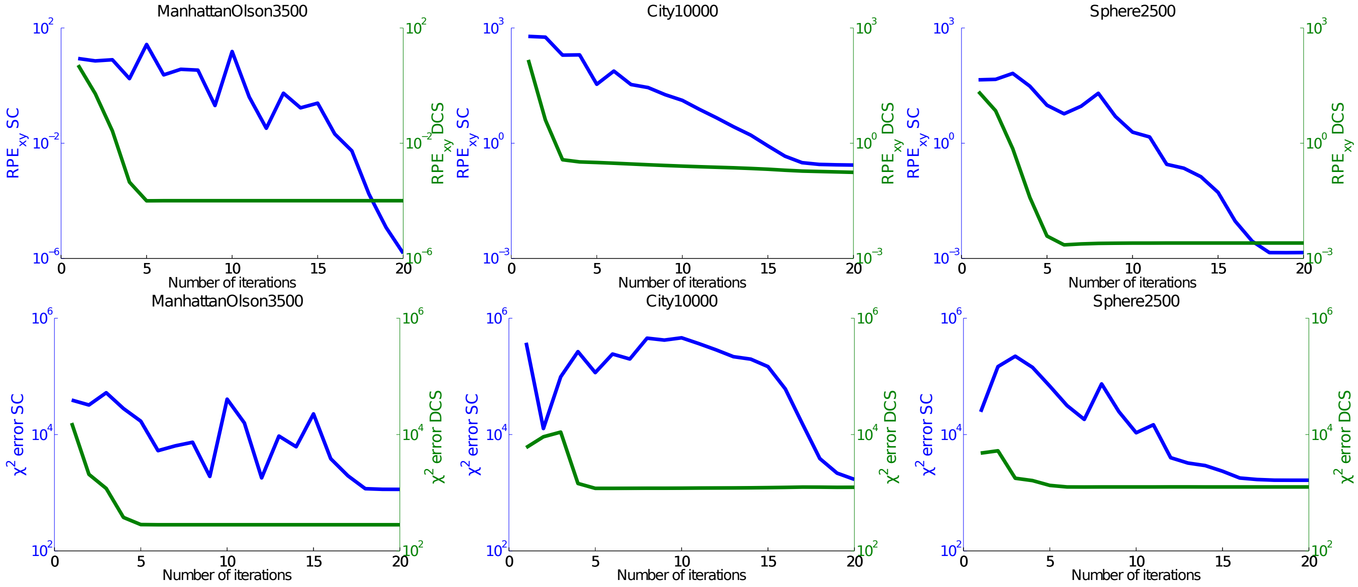


Fig. 8. The figure plots RPE (top row) and χ^2 error (bottom row) for 20 iterations for SC and DCS. While DCS converges within 6 iterations or less, SC needs between 15 and 20 iterations to converge. The shapes of the plots for SC reveal a frequent increase of RPE and χ^2 error which tend to indicate that there are more local minima in the SC formulation compared to DCS.

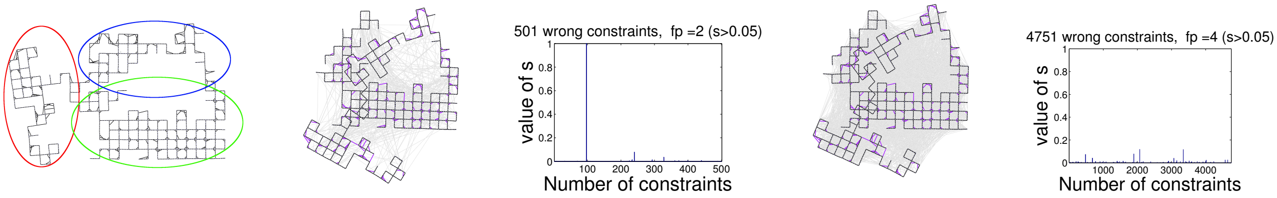


Fig. 9. Left: Ground truth configuration for Manhattan3500. The dataset reveals three sparsely connected region illustrated by the colored ellipses. The other four images are designed to illustrate the two failure cases, obtained for 501 and 4,751 random outliers, in the ManhattanOlson dataset. The images show the local minima maps in both situations together with the scaling values for the false positive constraints. The plots show that even if our method fails to converge to the optimal solution, the number of false positives accepted by the system is small, evident by a small scaling factor. With 501 outliers only two constraints have a scale value of more than 0.05 and with 4,751 outliers only four outliers have a scale value more than 0.05.

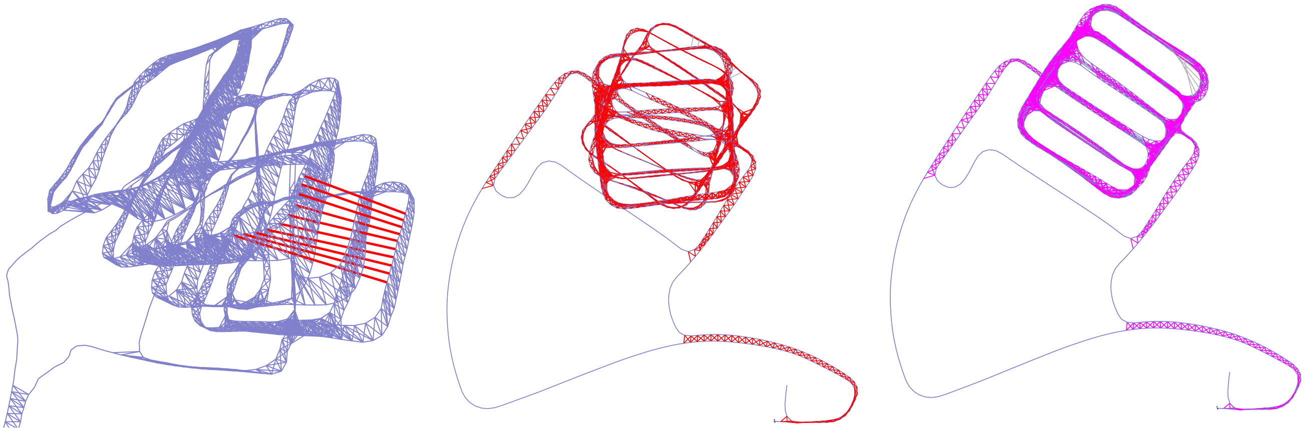


Fig. 10. Parking garage dataset with sparse connection. (Left) The original datasets with wrong loop closures connecting different decks in red. Note: the z-axis is scaled up to clearly show the wrong edges. (Center) SC returns the wrong solution while DCS rejects the outliers(right). This figure shows DCS being able to reject outliers even in the challenging case of datasets with minimal graph connectivity.

bad initialization in graph-based SLAM systems. We showed that DCS generalizes the switchable constraint method of Sünderhauf and Protzel [21], while introducing a substantially lower computational overhead. This is achieved by analyzing the behavior of the error function and deriving an analytical solution for computing the weighting factors. We implemented and thoroughly evaluated our approach. We supported our claims with extensive experiments and comparisons to state-of-the-art methods on publicly available datasets. The results show a comparable robustness to outliers as well as accuracy but with a convergence rate that is substantially faster. The authors have released the source code of the approach presented in this paper with the latest version of g2o.

VI. Acknowledgement

We thank E. Olson for the manhattanWorld dataset, Y. Latif for providing the processed Bicocca and Lincoln Lab datasets, E. Nebot and A. Ranganathan for original and processed Victoria Park datasets, M. Kaess for City10000, CityTrees10000, and Sphere2500 datasets, N. Sünderhauf for the Vertigo package and his open source implementation of switchable constraints.

References

[1] P. Agarwal, G.D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard. Robust map optimization using dynamic covariance scaling. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2013.

[2] M. Bosse, P. M. Newman, J. J. Leonard, and S. Teller. An ATLAS framework for scalable mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2003.

[3] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Int. Journal of Robotics Research*, 27(6), 2008.

[4] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Int. Journal of Robotics Research*, 25(12):1181–1204, 2006.

[5] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2), 2005.

[6] G. Grisetti, R. Kümmerle, and K. Ni. Robust optimization of factor graphs by using condensed measurements. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[7] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.

[8] G. Grisetti, C. Stachniss, and W. Burgard. Non-linear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 2009.

[9] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Int. Symp. on Comput. Intell. in Rob. and Aut. (CIRA)*, 1999.

[10] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 2011.

[11] A. Howard, M.J. Matarić, and G. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2001.

[12] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast incremental smoothing and mapping with efficient data association. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007.

[13] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[14] Y. Latif, C. Cadena, and J. Neira. Robust loop closing over time. *Proc. of Robotics: Science and Systems (RSS)*, 2012.

[15] Yasir Latif, César Cadena, and José Neira. Realizing, reversing, recovering: Incremental robust loop closing over time using the irrr algorithm. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4211–4217. IEEE, 2012.

[16] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4, 1997.

[17] E. Olson. Recognizing places using spectrally clustered local matches. *Robotics and Autonomous Systems*, 2009.

[18] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. In *Proc. of Robotics: Science and Systems (RSS)*, 2012.

[19] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2006.

[20] N. Sünderhauf. Vertigo: Versatile extensions for robust inference using graphical models, 2012.

[21] N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph slam. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[22] G.D. Tipaldi, M. Braun, and K.O. Arras. FLIRT: Interest regions for 2D range data with applications to robot navigation. In *Proc. of the Int. Symposium on Experimental Robotics (ISER)*, 2010.

[23] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15(1):59–76, 1997.